**Project 1: *Four Jira Enhancements for Vault and IDS Web Platform***

* **The Task**
  + I had four tasks that included creating update SQL scripts and changing the HTML code for the specific files in the platforms
* **What I learned**
  + The tasks may have been small, but they taught me a lot. For example:
    - Commit and Push my code to a repository that is used by more than one person (in comparison to when I use GitHub in my personal projects)
    - Git is not the only revision control system. There is Apache Subversion (A.K.A SVN)
    - Understanding what the requirements are asking me to do

**Project 2: *Learning about Containers, Open Source Technologies, and Microservices***

* **The Task**
  + Learn about Containers, Open Source Technologies (Docker), Microservices
  + Run a simple web application on a container
  + Make a presentation on what I have learned
* **What I learned**
  + Container: a unit that contains everything needed to run the application
    - ***Check out my presentation to learn more about Docker and Containers***
  + Project open doors to learning about Web Applications, Cloud Services, Microservices, Open Source Technology

**Project 3: *Web Applications with Flask (A Python Microframework)***

* **The Task**
  + For better understanding of Docker Tutorials,

I decided to learn Flask

* + Required to run a simple web application on a container
  + **What I learned**
  + What a web application is
  + Client Side and Server Side
  + Used Python Flask as the Server Side Language
    - ***Link:*** [***http://flask-app-practice.appspot.com/***](http://flask-app-practice.appspot.com/)
    - ***Link:*** [***http://spgi2018-container-project.appspot.com/***](http://spgi2018-container-project.appspot.com/)

**Project 4: *Playing with Cloud Services***

* **The Task**
  + After running the two web applications in a container, I wanted to show others the projects
  + Cloud services allow me to deploy my web application
  + **What I learned**
  + Cloud Services like: Heroku, AWS, Google App Engine, PythonAnywhere
  + Different process each one takes to deploy a web application
  + Difference between to GitHub Pages and Cloud Services

**Project 5: *Creating a RESTful Client and Using APIs***

* **The Task**
  + **Part 1:**
    - Create an enhancement for CARE web platform that allow users to check if file is locked or not
    - Use Box RESTful API, but there was not RESTful service to check if a file is locked or not
    - Tasked with creating a RESTful client that checks if a file is locked or not
  + **Part 2:**
    - Create a RESTful client that allows developers to know if there is an error with the file.
* **What I learned**
  + Application Programming Interface (API)
    - API part of the server that receives requests and send responses
    - A company offers an API it means that they have built a set of URLS that return data
  + RESTful APIs
    - APIs that conform with REST architectural Style
  + JavaScript Object Notation (Json)
    - Lightweight data-interchange format
  + Java Spring Framework
    - Application development framework for Java.
    - Used to make RESTful client
  + Postman
    - A tool to test REST methods on RESTful APIs